Makeup Quiz

Problem 1: Master Theorem and Recurrence Relations

Recurrence: $T(n) = 4T(n/2) + n^2$.

Identifying parameters: $a = 4$, $b = 2$, $f(n) = n^2$.
Computing critical exponent: $\log_b a = \log_2 4 = 2$, so $n^{\log_b a} = n^2$.

We have $f(n) = \theta(n^{\log_b a})$. This matches Master Theorem case 2 (the balanced case): if $f(n) = \theta(n^{\log_b a} \log^k n)$with $k = 0$, then

$$T(n) = \theta(n^{\log_b a} \log^{k+1} n) = \theta(n^2 \log n).$$

Answer (a): $T(n) = \theta(n^2 \log n)$.
Answer (b): Because $f(n) = n^2$equals $n^{\log_b a}$exactly, the recurrence falls in Master Theorem case 2; hence the extra $\log n$factor appears.

Problem 2: Prove Why the Modified Bellman Ford's Algorithm Fails

(a) Why standard Bellman - Ford needs $|V| - 1$passes

Any simple path has at most $|V| - 1$edges. Each full pass of relaxing every edge can increase by at most one the maximum number of edges of shortest paths whose distances have been correctly propagated from the source. By induction on the number of passes: after $k$passes, all shortest paths that use at most $k$edges have their distances correctly computed. Thus $|V| - 1$passes suffice to settle all simple shortest paths (assuming no negative cycles).

(b) Why the single-pass modified algorithm can fail

A single pass relaxes each edge once in some fixed order. If an improved distance for an intermediate vertex is discovered after the relaxation of an outgoing edge from that intermediate, that improvement will not propagate further in the same single pass. Therefore distances can remain too large even with non-negative weights.

(c) Counterexample (non-negative weights)

Graph $G' = (V', E')$with

- $V' = \{s, u, v\}$,
- edges and weights: (s, u)weight 1, (u, v)weight 1, (s, v)weight 5.

Assume the single-pass edge processing order: $(s, v), (u, v), (s, u)$.

Initial distances: $d[s] = 0, d[u] = d[v] = +\infty$.

Relaxations in order:

1. (s, v): relax sets $d[v] = 5$.
2. (u, v): $d[u] = +\infty$ so no change.
3. (s, u): relax sets $d[u] = 1$.

End of pass: $d[s] = 0$, $\mathrm{d}[u] = 1$, $\mathrm{d}[v] = 5$.

True shortest-path distances from s:

- $\mathrm{dist}\ (s) = 0$,
- $\mathrm{dist}\ (u) = 1$,
- $\mathrm{dist}\ (v) = 2$ via $s \to u \to v$.

Difference: The algorithm returns $d[v] = 5$ but the true shortest distance is 2. The failure occurs because the improvement to u happened after (u, v) was processed.

(d) Why $|V| - 1$ passes are necessary and sufficient

- Sufficient: Any simple shortest path has at most $|V| - 1$ edges. Repeating relaxation for $|V| - 1$ passes propagates distance improvements along every edge of such a path; by induction every shortest distance is settled by pass $|V| - 1$.
- Necessary (in the worst case): If you use fewer than $|V| - 1$passes, you cannot guarantee discovering shortest paths that require more edges than the number of passes. A chain $s \to v_1 \to v_2 \to \cdots \to v_k$of length k needs at least k passes to propagate the source distance to the last vertex, so fewer passes can fail in such cases.
- These statements assume there are no negative-weight cycles; if such cycles exist, no finite number of passes yields well-defined finite shortest-path distances.